UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/437,169	11/10/1999	MARK D. RUSTAD	977.029US1	7612

21186 7590 09/18/2003

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.
P.O. BOX 2938
MINNEAPOLIS, MN 55402

EXAMINER
KING, JUSTIN

ART UNIT	PAPER NUMBER
2181	10

DATE MAILED: 09/18/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/437,169	RUSTAD, MARK D.	
	Examin r	Art Unit	
	Justin I. King	2181	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 July 2003* .

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5,8,11-20,23-31,34-38,41-46,49,52,55-64 and 67-77* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-5,8,11-20,23-31,34-38,41-46,49,52,55-64 and 67-77* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

3. Claims 1-5, 8, 11, 14-20, 26-31, 34, 36-38, 41-46, 49, 52, 55, 58-59, 60-64, and 70-77 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lange et al. (U.S. Patent No. 3,845,474) in view of Averill (U.S. Patent No. 5,313,591).

Referring to claim 1: Lange discloses a system including a bus, a resource, and a plurality of entities (figure 1), and at least one entity among the plurality of entities including a memory (figure 1, structures 11). Lange does not explicitly state resetting the current owner's memory when the current owner is a different entity from the previous owner.

Averill teaches resetting the current owner's memory when the current owner is a different entity from the previous owner. Averill discloses that it is known to keep each

processor's cache accurate and current (column 1, lines 25-26), and purge or flush must be done before any new processor gains control of the bus (column 1, lines 23-42). Averill's system recognizes the processor with gaining new control and resetting (purge or flush) its cache accordingly, which is discriminating between consecutive and disparate data owners in determining whether to reset a portion of memory.

Hence, it would have been obvious to one having ordinary skill in the computer art to adapt the Averill's teaching into Lange because Averill teaches one to achieve the multiprocessor system's cache coherency by flushing or purging each processor's cache.

Referring to claim 2: The claim 2 is rejected over Lange in view of Averill as stated above; furthermore, Lange's entity is an integrated circuit (column 4, lines 2-8).

Referring to claim 3: The claim 3 is rejected over Lange in view of Averill as stated above; furthermore, Lange's resource is a main storage, which is a memory device.

Referring to claim 4: The claim 4 is rejected over Lange in view of Averill as stated above; furthermore, Lange discloses a manager (figure1, structure 3, column 1, lines 44-46, column 3, lines 40-43) to manage the access to the resource.

Referring to claim 5: The claim 5 is rejected over Lange in view of Averill as stated above; furthermore, Averill discloses that an arbitration protocol to arbitrate the shared bus among processors (column 1, lines 17-22), and teaches one on how to arbitrate the shared bus (column 2, lines 40-52). Averill's arbitration protocol is the arbiter. Hence, it would have been obvious to one having ordinary skill in the computer art to adapt Averill's teaching into Lange because Averill teaches one on how to arbitrate the shared bus when multiple processors simultaneously contend for control.

Referring to claim 8: Lange discloses a bus, a resource (figure 1, structure main store), a plurality of processors including a first and second processors (figure 1, processors 1-2), the first processor includes a fast memory (figure 1, structures 11), and a switching mechanism (figure 1, structure 3). Lange does not reset the memory of the previous resource controlling entity when the resource control is switched out to a different entity.

Averill teaches that the previously resource controlling entity's memory is reset while the second entity gains the resource control (column 1, lines 34-40), and Averill's previous resource controlling entity's memory resetting operation at the time of the giving away the resource control (purge or flush).

Hence, it would have been obvious to one having ordinary skill in the computer art to adapt the Averill's teaching into Lange because Averill teaches one to achieve the multiprocessor system's cache coherency by flushing or purging each processor's cache.

Referring to claim 11: Claim 8's argument applies; furthermore, Lange's switch mechanism is a hardware device.

Referring to claims 14-15: Claim 8's argument applies; furthermore, Lange's shared resource is a memory, which is a hardware device.

Referring to claim 16: Claim 8's argument applies; furthermore, Lange discloses a communication channel controller (figure 1, structures 3 and 5).

Referring to claim 17: Claim 8's argument applies; furthermore, the data stored in Lange's shared resource is the software resource.

Referring to claim 18: Claims 8 and 17's arguments apply; furthermore, in order to properly store and retrieve the data, the memory has a particular file system structure, such as the FAT or FAT16, such that, the stored software data source is a data structure.

Referring to claim 19: Claim 8's argument applies; furthermore, Lange's fast memory (figure 1, structure 11) is a cache memory.

Referring to claim 20: Lange discloses a bus, a resource (figure 1, structure main store), and a plurality of processors including a first and second processors (figure 1, processors 1-2), the first processor includes a fast memory (figure 1, structures 11), and a switch mechanism providing exclusive control (figure 1, structure 3, column 1, lines 44-46, column 3, lines 40-43). Lange does not reset the memory of the previous resource controlling entity when the resource control is switched out to a different entity.

Averill teaches that the previously resource controlling entity's memory is reset while the second entity gains the resource control (column 1, lines 34-40), and Averill's previous resource controlling entity's memory resetting operation at the time of the giving away the resource control (purge and flush).

Hence, it would have been obvious to one having ordinary skill in the computer art to adapt the Averill's teaching into Lange because Averill teaches one to achieve the multiprocessor system's cache coherency by flushing or purging each processor's cache.

Referring to claim 26: Claim 20's argument applies; furthermore, claim 16's argument applies.

Referring to claim 27: Claim 20's argument applies; furthermore, Lange's fast memory (figure 1, structure 11) is cache memory.

Referring to claim 28: Claims 20 and 27's argument applies; furthermore, Lange does not specify the cache memory as a primary cache (L1 cache); L1 cache is well-known industrial practice.

Referring to claim 29: Claim 20's argument applies; furthermore, Lange discloses that entity has a cache, although Lange does not specify particular types of cache, such as a secondary cache (L2 cache); every cache type is functioning at the same purpose as to store local information for speedy processing; the type of cache in which each processor uses is inconsequential for the invention as a whole and presents no new or unexpected results, so long as the data on the cache is reset.

Referring to claim 30: Lange discloses a bus, a resource (figure 1, structure main store), and a plurality of processors including a first and second processors (figure 1, processors 1-2), the first processor includes a fast memory (figure 1, structures 11), and a switch mechanism providing exclusive control (figure 1, structure 3, column 1, lines 44-46, column 3, lines 40-43). Lange does not explicitly disclose the data structure and the indicating state.

Averill teaches that the previously resource controlling entity's memory is reset while the second entity gains the resource control (column 1, lines 34-40), and Averill's previous resource controlling entity's memory resetting operation at the time of the giving away the resource control (purge and flush). Such Averill's means for identifying the previous controlling entity and the current controlling entity is the first and the second identifiers, and trigger for purge and flush is the state for indicating the resource is under control.

Hence, it would have been obvious to one having ordinary skill in the computer art to adapt the Averill's teaching into Lange because Averill teaches one to achieve the multiprocessor system's cache coherency by flushing or purging each processor's cache.

Referring to claim 31: Averill teaches that the previously resource controlling entity's memory is reset while the second entity gains the resource control (column 1, lines 34-40), and Averill's previous resource controlling entity's memory resetting operation at the time of the giving away the resource control (purge and flush).

Referring to claim 34: Claim 19's argument applies.

Referring to claim 36: Lange teaches that the data structure on cache including at least one location of at least one portion of the resource (figure 3).

Referring to claim 37: Lange discloses a bus, a resource (figure 1, structure main store), and a plurality of processors including a first and second processors (figure 1, processors 1-2), the first processor includes a fast memory (figure 1, structures 11), and a switch mechanism providing exclusive control (figure 1, structure 3, column 1, lines 44-46, column 3, lines 40-43). Lange does not explicitly disclose the data structure and the indicating state.

Averill teaches that the previously resource controlling entity's memory is reset while the second entity gains the resource control (column 1, lines 34-40), and Averill's previous resource controlling entity's memory resetting operation at the time of the giving away the resource control (purge and flush). Such Averill's means for identifying the previous controlling entity and the current controlling entity is the first and the second identifiers, and trigger for purge and flush is the state for indicating the resource is under control.

Hence, it would have been obvious to one having ordinary skill in the computer art to adapt the Averill's teaching into Lange because Averill teaches one to achieve the multiprocessor system's cache coherency by flushing or purging each processor's cache.

Referring to claim 38: Claim 37's argument applies; further, claim 19's argument applies.

Referring to claim 41: Claim 37's argument applies; furthermore, the resource distribution must be done according to a particular orderly fashion, and the FIFO is a basic algorithm for handling transaction request.

Referring to claim 42: Lange discloses a bus, a resource (figure 1, structure main store), and a plurality of processors including a first and second processors (figure 1, processors 1-2), the first processor includes a fast memory (figure 1, structures 11), and a switch mechanism providing exclusive control (figure 1, structure 3, column 1, lines 44-46, column 3, lines 40-43). Lange does not explicitly disclose the data structure and the indicating state.

Averill teaches that the previously resource controlling entity's memory is reset while the second entity gains the resource control (column 1, lines 34-40), and Averill's previous resource controlling entity's memory resetting operation at the time of the giving away the resource control (purge and flush). Such Averill's means for identifying the previous controlling entity and the current controlling entity is the first and the second identifiers, and trigger for purge and flush is the state for indicating the resource is under control.

Hence, it would have been obvious to one having ordinary skill in the computer art to adapt the Averill's teaching into Lange because Averill teaches one to achieve the multiprocessor system's cache coherency by flushing or purging each processor's cache.

Referring to claim 43: Lange discloses a bus, a resource (figure 1, structure main store), and a plurality of processors including a first and second processors (figure 1, processors 1-2), the first processor includes a fast memory (figure 1, structures 11), and a switch mechanism providing exclusive control (figure 1, structure 3, column 1, lines 44-46, column 3, lines 40-43). Lange does not reset the memory of the previous resource controlling entity when the resource control is switched out to a different entity.

Averill teaches that the previously resource controlling entity's memory is reset while the second entity gains the resource control (column 1, lines 34-40), and Averill's previous resource controlling entity's memory resetting operation at the time of the giving away the resource control (purge and flush).

Hence, it would have been obvious to one having ordinary skill in the computer art to adapt the Averill's teaching into Lange because Averill teaches one to achieve the multiprocessor system's cache coherency by flushing or purging each processor's cache.

Referring to claim 44: Claim 43's argument applies; furthermore, Lange's fast memory is a cache memory.

Referring to claim 45: Claim 43's argument applies; furthermore, Lange's system controller (figure 1, structure 3) is the communication channel controller receptive to diverse communication protocols (protocols from structures 4, 5, and processors 1-2).

Referring to claim 46: Claim 43's argument applies; furthermore, claim 16's argument applies.

Referring to claim 49: Lange discloses a bus, a resource (figure 1, structure main store), a plurality of processors including a first and second processors (figure 1, processors 1-2), the first

processor includes a fast memory (figure 1, structures 11), and a scheduler (figure 1, structure 3). Lange does not reset the memory of the previous resource controlling entity when the resource control is switched out to a different entity.

Averill teaches that the previously resource controlling entity's memory is reset while the second entity gains the resource control (column 1, lines 34-40), and Averill's previous resource controlling entity's memory resetting operation at the time of the giving away the resource control (purge or flush).

Hence, it would have been obvious to one having ordinary skill in the computer art to adapt the Averill's teaching into Lange because Averill teaches one to achieve the multiprocessor system's cache coherency by flushing or purging each processor's cache.

Referring to claim 52: Lange discloses a bus, a resource (figure 1, structure main store), a plurality of processors including a first and second processors (figure 1, processors 1-2), the first processor includes a fast memory (figure 1, structures 11), and a switching mechanism (figure 1, structure 3). Lange does not reset the memory of the previous resource controlling entity when the resource control is switched out to a different entity.

Averill teaches that the previously resource controlling entity's memory is reset while the second entity gains the resource control (column 1, lines 34-40), and Averill's previous resource controlling entity's memory resetting operation at the time of the giving away the resource control (purge or flush).

Hence, it would have been obvious to one having ordinary skill in the computer art to adapt the Averill's teaching into Lange because Averill teaches one to achieve the multiprocessor system's cache coherency by flushing or purging each processor's cache.

Referring to claim 55: Claim 52's argument applies; furthermore, Lange's switch mechanism is a hardware device.

Referring to claims 58-59: Claim 52's argument applies; furthermore, Lange's shared resource is a memory, which is a hardware device.

Referring to claim 60: Claim 52's argument applies; furthermore, claim 16's argument applies.

Referring to claim 61: Claim 52's argument applies; furthermore, the data stored in Lange's shared resource is the software resource.

Referring to claim 62: Claims 52 and 61's arguments apply; furthermore, in order to properly store and retrieve the data, the memory has a particular file system structure, such as the FAT or FAT16, such that, the stored software data source is a data structure.

Referring to claim 63: Claim 52's argument applies; furthermore, Lange's fast memory (figure 1, structure 11) is a cache memory.

Referring to claim 64: Lange discloses a bus, a resource (figure 1, structure main store), and a plurality of processors including a first and second processors (figure 1, processors 1-2), the first processor includes a fast memory (figure 1, structures 11), and a switch mechanism providing exclusive control (figure 1, structure 3, column 1, lines 44-46, column 3, lines 40-43). Lange does not reset the memory of the previous resource controlling entity when the resource control is switched out to a different entity.

Averill teaches that the previously resource controlling entity's memory is reset while the second entity gains the resource control (column 1, lines 34-40), and Averill's previous resource

controlling entity's memory resetting operation at the time of the giving away the resource control (purge and flush).

Hence, it would have been obvious to one having ordinary skill in the computer art to adapt the Averill's teaching into Lange because Averill teaches one to achieve the multiprocessor system's cache coherency by flushing or purging each processor's cache.

Referring to claim 70: Claim 64's argument applies, furthermore, claim 16's argument apples.

Referring to claim 71: Claim 64's argument applies; furthermore, Lange's fast memory (figure 1, structure 11) is cache memory.

Referring to claims 72-73: Claims 64 and 71's argument applies; furthermore, Lange does not specify the cache memory as a primary cache (L1 cache). Every cache type is functioning at the same purpose as to store local information for speedy processing; the type of cache in which each processor uses is inconsequential for the invention as a whole and presents no new or unexpected results, so long as the data on the cache is reset.

Referring to claim 74: Lange discloses a bus, a resource (figure 1, structure main store), and a plurality of processors including a first and second processors (figure 1, processors 1-2), the first processor includes a fast memory (figure 1, structures 11), and a switch mechanism providing exclusive control (figure 1, structure 3, column 1, lines 44-46, column 3, lines 40-43). Lange does not reset the memory of the previous resource controlling entity when the resource control is switched out to a different entity.

Averill teaches that the previously resource controlling entity's memory is reset while the second entity gains the resource control (column 1, lines 34-40), and Averill's previous resource

controlling entity's memory resetting operation at the time of the giving away the resource control (purge and flush).

Hence, it would have been obvious to one having ordinary skill in the computer art to adapt the Averill's teaching into Lange because Averill teaches one to achieve the multiprocessor system's cache coherency by flushing or purging each processor's cache.

Referring to claim 75: Claim 74's argument applies; furthermore, Lange's fast memory is a cache memory.

Referring to claim 76: Claim 74's argument applies; furthermore, Lange's system controller (figure 1, structure 3) is the communication channel controller receptive to diverse communication protocols (protocols from structures 4, 5, and processors 1-2). Lange does not disclose a processor coupled to the communication channel.

Referring to claim 77: Claim 74's argument applies; furthermore, Lange teaches the lock and switch for the cache coherency (figure 1, structure 3).

4. Claims 12, 23-25, 56, 67-69 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lange in view of Averill, and in further view of the Structured Computer Organization by Andrew Tanenbaum.

Referring to claim 12: Claim 8's argument applies; furthermore, none of Lange or Averill discloses a software switch. Tanenbaum discloses that the hardware and software are logically equivalent, and Tanenbaum further discloses that the decision to put certain functions in hardware or software is based on factors as cost, speed, reliability, and frequency of expected changes. Hence, it would have been obvious to one having ordinary skill in the computer art at

the time Applicant made the invention to adapt Tanenbaum's teaching into Lange and Averill because Tanenbaum teaches one to interchange software and hardware based on the cost, speed, reliability, and frequency of expected changes.

Referring to claims 23-24: Claim 20's argument applies; furthermore, Tanenbaum discloses that the hardware and software are logically equivalent, and Tanenbaum further discloses that the decision to put certain functions in hardware or software is based on factors as cost, speed, reliability, and frequency of expected changes. Hence, it would have been obvious to one having ordinary skill in the computer art at the time Applicant made the invention to adapt Tanenbaum's teaching into Lange and Averill because Tanenbaum teaches one to interchange software and hardware based on the cost, speed, reliability, and frequency of expected changes.

Referring to claim 25: Claims 20 and 24's arguments apply; furthermore, a binary object is generated after the software source code is compiled.

Referring to claim 56: Claim 52's argument applies; furthermore, neither Lange nor Averill discloses a software switch. Tanenbaum discloses that the hardware and software are logically equivalent, and Tanenbaum further discloses that the decision to put certain functions in hardware or software is based on factors as cost, speed, reliability, and frequency of expected changes. Hence, it would have been obvious to one having ordinary skill in the computer art at the time Applicant made the invention to adapt Tanenbaum's teaching into Lange and Averill because Tanenbaum teaches one to interchange software and hardware based on the cost, speed, reliability, and frequency of expected changes.

Referring to claims 67-68: Claim 64's argument applies; furthermore, Lange and Averill do not specify whether the lock is a software or hardware. Tanenbaum discloses that the

hardware and software are logically equivalent, and Tanenbaum further discloses that the decision to put certain functions in hardware or software is based on factors as cost, speed, reliability, and frequency of expected changes. Hence, it would have been obvious to one having ordinary skill in the computer art at the time Applicant made the invention to adapt Tanenbaum's teaching into Lange and Averill because Tanenbaum teaches one to interchange software and hardware based on the cost, speed, reliability, and frequency of expected changes.

Referring to claim 69: Claim 64's argument applies; furthermore, a binary object is generated after the software source code is compiled.

5. Claims 13 and 57 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lange in view of Averill, and in further view of Tanenbaum and Georgiou et al. (U.S. Patent No. 4,633,394).

Referring to claim 13: Claims 8 and 12's arguments apply; furthermore, none of Lange, Averill, or Tanenbaum discloses that the switch is a Djikstra primitive. Georgiou teaches implementing distributed arbitration among multiple processors and shared resource with Djikstra primitive. Hence, it would have been obvious to one having ordinary skill in the computer art to adapt Georgiou's teaching into Lange, Averill, and Tanenbaum because Georgiou teaches one to route data with the Djikstra primitive to achieve the shortest path from the source to the destination.

Referring to claim 57: Claims 52 and 56's arguments apply; furthermore, none of Lange, Averill, or Tanenbaum discloses that the switch is a Djikstra primitive. Georgiou teaches implementing distributed arbitration among multiple processors and shared resource with

Djikstra primitive. Hence, it would have been obvious to one having ordinary skill in the computer art to adapt Georgiou's teaching into Lange, Averill, and Tanenbaum because Georgiou teaches one to route data with the Djikstra primitive to achieve the shortest path from the source to the destination.

6. Claims 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over the combination of the Lange, Averill, and Gusefski (U.S. Patent No. 5,202,972).

Referring to claim 35: Gusefski discloses a plurality of processors and shared resources (figure 2) and the processor has a cache memory (figure 2, structure 18). Gusefski further discloses an exclusive control over the shared resource to only one of the requesting processors (column 6, lines 11-12), and a bus switch unit for scheduling the resource access (column 5, lines 38-40). Hence, it would have been obvious to one having ordinary skill in the computer art to adopt Gusefski's teaching to Lange and Averill because Gusefski teaches one on how to achieve the cache coherence only with needed cache portion.

Response to Arguments

7. In response to that Applicant claims the system described by Gusefski requires that each processor monitors for fetch conflicts, and quotes figure 6 and column 14, lines 23-58 (Remark, page 14, response to the 102 and 103 Rejection, 3rd paragraph's last 2 lines, page 16, 3rd paragraph, page 17, 3rd paragraph's last 3 lines): Examiner reviews the quoted section of

Gusefski, and find no support to Applicant's claim on that Gusefski's each processor monitors for fetch conflicts.

8. In response to Applicant's argument that Averill does not teach a system which discriminates between consecutive and disparate data owners in determining whether to reset a portion of memory (Remark, page 15, last 3 lines, page 16, 4th paragraph's last 3 lines, page 17, 2nd paragraph, page 18, 5th paragraph's first 3 lines): Averill does teach this. Averill discloses that it is known to keep each processor's cache accurate and current (column 1, lines 25-26), and purge or flush must be done before any new processor gains control of the bus (column 1, lines 23-42). Averill's system recognizes the processor with gaining new control and resetting its cache accordingly, which is discriminating between consecutive and disparate data owners in determining whether to reset a portion of memory.

9. In response to Applicant's statement that claims 37-38, 41, and 42 teach that it is possible to place a first identifier and a second identifier in the resource controller (Remark, page 17, last paragraph): The claims' language does not explicitly place a first identifier and a second identifier *in the resource controller.*

Conclusion

10. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Justin King whose telephone number is (703) 305-4571. The examiner can normally be reached on Monday through Friday from 9:00 A.M. to 5:00 P.M..

If attempts to reach the examiner by telephones are unsuccessfully, the examiner's supervisor, Mark Reinhart can be reached at (703) 308-3110.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose number is (703)-306-5631.

Justin King
September 4, 2003

Gopal C. Ray
GOPAL C. RAY
PRIMARY EXAMINER
GROUP 2100